Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Improved deep learning aided key recovery framework: applications to large-state block ciphers*#

Xiaowei LI, Jiongjiong REN†‡, Shaozhen CHEN

School of Cyber Science and Technology, Information Engineering University, Zhengzhou 450000, China

†E-mail: jiongjiong_fun@163.com

Received Dec. 19, 2023; Revision accepted Mar. 19, 2024; Crosschecked Sept. 4, 2024

Abstract: At the Annual International Cryptology Conference in 2019, Gohr introduced a deep learning based cryptanalysis technique applicable to the reduced-round lightweight block ciphers with a short block of SPECK32/64. One significant challenge left unstudied by Gohr's work is the implementation of key recovery attacks on large-state block ciphers based on deep learning. The purpose of this paper is to present an improved deep learning based framework for recovering keys for large-state block ciphers. First, we propose a key bit sensitivity test (KBST) based on deep learning to divide the key space objectively. Second, we propose a new method for constructing neural distinguisher combinations to improve a deep learning based key recovery framework for large-state block ciphers and demonstrate its rationality and effectiveness from the perspective of cryptanalysis. Under the improved key recovery framework, we train an efficient neural distinguisher combination for each large-state member of SIMON and SPECK and finally carry out a practical key recovery attack on the large-state members of SIMON and SPECK. Furthermore, we propose that the 13-round SIMON64 attack is the most effective approach for practical key recovery to date. Noteworthly, this is the first attempt to propose deep learning based practical key recovery attacks on 18-round SIMON128, 19-round SIMON128, 14-round SIMON96, and 14-round SIMON64. Additionally, we enhance the outcomes of the practical key recovery attack on SPECK large-state members, which amplifies the success rate of the key recovery attack in comparison to existing results.

Key words: Deep learning; Large-state block cipher; Key recovery; Differential cryptanalysis; SIMON; SPECK
https://doi.org/10.1631/FITEE.2300848 **CLC number:** TN918; TP18

1 Introduction

Differential analysis (Biham and Shamir, 1993) is a highly effective technique for attacking iterative block ciphers. Its fundamental concept involves the recovery of certain key bits by analyzing the effect of the differences between plaintext and ciphertext pairs. With the rapid development of artificial intelligence, it is worth mentioning that the combination of deep learning and cryptanalysis has emerged as a trendy area of research in modern cryptography. One notable contribution is Gohr's research presented at the Annual International Cryptology Conference in 2019 (Crypto'19), which holds great importance.

Gohr (2019) introduced a cryptanalysis technique based on deep learning at the Crypto'19. The method was applied to attack the reduced-round lightweight block cipher SPECK32/64 (Beaulieu et al., 2015). Gohr constructed a neural distinguisher for SPECK32/64 using a deep neural network to differentiate real pairs from random pairs and successfully performed the key recovery attack for SPECK32/64. When compared to conventional differential cryptanalysis, the deep learning based

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 62206312)
# Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2300848) contains supplementary materials, which are available to authorized users
ORCID: Jiongjiong REN, https://orcid.org/0000-0003-2223-4329

key recovery attack exhibits lower complexity.

Following Gohr's work, Baksi (2022) pioneered a deep learning based construction approach for non-Markovian cipher distinguishers, resulting in distinguishers for both Gimli-Hash and Gimli-Cipher. Bellini and Rossi (2021) used both multilayer perceptron and convolutional neural networks to create neural network distinguishers for the reduced-round tiny encryption algorithm (TEA) and its evolution RAIDEN. They presented the occasions when traditional distinguishers cannot be applied and the limitations of the approach. Jain et al. (2020) developed a multilayer perceptron neural network distinguisher to analyze lightweight block cipher PRESENT, which can distinguish with high probability between 3–6 rounds of PRESENT algorithmic cipher data and randomly generated data. Chen and Yu (2021) developed a novel neural network differential distinguisher model that applies multiple ciphertext pairs, instead of a single pair, to enhance the neural distinguisher for 5–7 rounds of SPECK32. Zhang et al. (2022) improved the accuracy of 5–8 rounds of the SPECK32 algorithm distinguishers by using an inception block consisting of multiple parallel convolutional layers before a residual network. These novel approaches extended the application of deep learning to the cryptanalysis of block ciphers.

Key recovery attacks based on deep learning have made some progress in recent years. Chen et al. (2023) found that the key recovery attack proposed by Gohr relies heavily on practical experiments and cannot be used for theoretical analysis. To overcome this limitation, they introduced the idea of building a neural distinguisher on partial ciphertext bits, based on the fact that only partial ciphertext bits have an effect on the neural distinguishers, to reduce the key space. Additionally, Gohr's method for recovering the key can be employed but a sufficiently large quantity of neutral bits in the pre-differential are necessary. Bao et al. (2022) broadened the concept of neutral bits to find more neutral bits and increase the number of attack rounds by using the pre-differentials. Chen et al. (2022) proposed a deep learning aided key recovery framework for large-state block ciphers and tested it on large-state SPECK. This creates a new possibility for implementing deep learning aided key recovery attacks on large-state block ciphers. However, the following issues still exist and deserve further exploration and research:

How can the key recovery framework for large-state block ciphers be refined to apply key recovery attacks on more algorithms? How can the structural features of various cryptographic algorithms be fully used to increase the number of attack rounds and improve key recovery results?

Considering that Gohr's key recovery attack requires guessing all bits of the round key at once, the attack complexity is excessively high. We discover that the key has a direct impact on the state of the ciphertext. Therefore, we propose a technique that tests the sensitivity of the key bits and then integrate it with the algorithm round function structure to reduce the key space. Additionally, we propose a new approach that includes constructing a combination of neural distinguishers and retraining the neural distinguisher with the bits that exert a greater influence. This is due to the fact that the neural distinguisher suggested by Gohr uses only a portion of the available bit knowledge (Chen et al., 2023), making it difficult to recover bits with minimal or without impact on the neural distinguisher. Furthermore, we improve the framework of the deep learning aided key recovery attack and validate our approach on large-state members of both SIMON and SPECK (Beaulieu et al., 2015). The main contributions of this paper are as follows:

1. Design a deep learning based key bit sensitivity test and divide the round key space. By using this technique, we aim to find an effective set of neural distinguishers that are trained with single-bit input differences, to achieve reduction of the round key. In this case, each neural distinguisher will correspond to a specific subset of the round key and will be sensitive to some of the round key bits. By concatenating these subsets of the round key, we will capture most, if not all, of the round key bits.

2. Propose a new method for constructing neural distinguisher combinations and improve a deep learning aided key recovery framework for large-state block ciphers. This approach guarantees the highest possible accuracy of the neural distinguisher built on partial ciphertext state bits while ensuring a feasible and balanced workload for recovering partial keys for each neural distinguisher. We create a neural distinguisher combination using the discovered neural distinguishers to improve the deep learning aided key recovery framework for large-state block ciphers. To confirm the method efficiency, we train an effective

neural distinguisher combination for each large-state member of SIMON and SPECK.

3. Propose a practical key recovery attack on large-state SIMON members (SIMON64, SIMON96, and SIMON128) and improve the results of the practical key recovery attack on large-state SPECK members (SPECK64, SPECK96, and SPECK128). We execute a practical key recovery attack on 13-round SIMON64. In comparison with a previous attack (Hou et al., 2023) on 13-round SIMON64, the key recovery attack of this paper significantly diminishes the time complexity and substantially enhances the success rate of key recovery. Additionally, it is the first to propose deep learning based practical key recovery attacks on 14-round SIMON64, 14-round SIMON96, 18-round SIMON128, and 19-round SIMON128. For the large-state SPECK, in comparison with the attack studied by Chen et al. (2022), we increase the success rate of key recovery and decrease the average Hamming distance between the guessed key and the correct round key hw(kg, rk). To the best of our knowledge, this paper presents the most effective deep learning based practical key recovery attacks on 13-round SIMON64 and large-state SPECK. Table 1 provides a comparison between the attacks presented here and those from the literature.

2 Preliminaries

2.1 Notations

The notations used in this paper are presented below:

SIMON$2n/mn$: SIMON acting on $2n$-bit plaintext blocks and using an mn-bit key;

SPECK$2n/mn$: SPECK acting on $2n$-bit plaintext blocks and using an mn-bit key;

(L_i, R_i): the left and right branches of a state after the encryption of i rounds;

rk_i: the subkey of the i^{th} round;

$\{i_2 \sim i_1\}$: the set of bit indices $\{i_2, i_2-1, \ldots, i_1\}$;

$\{[i_2] \sim [i_1]\}$: the set of neutral bits $\{[i_2], [i_2-1], \ldots, [i_1]\}$;

$\oplus$: bitwise XOR;

$\&$: bitwise AND;

$\odot$: addition modulo 2^n;

S^j : left circular shift by j bits;

S^{-j} : right circular shift by j bits;

K : master key;

P : plaintext;

C : ciphertext;

$\Delta_{[i]}$: a single-bit difference whose i^{th} bit is the only active bit;

hw(kg, rk): the Hamming distance between the guessed key kg and the round key rk.

2.2 General description of SIMON and SPECK

The National Security Agency (NSA) released the SPECK and SIMON algorithms simultaneously in 2013. The objective of SIMON (Beaulieu et al., 2015) is to create a block cipher suitable for hardware in resource-constrained environments, with a particular emphasis on flexibility. Consequently,

Table 1 Summary of practical key recovery attacks on large-state SIMON and SPECK

Block cipher	Number of rounds	Time complexity	Data complexity	Success rate	hw(kg, rk)	Work
SIMON128	18	$2^{19.94}$	$2^{16.58}$	0.81	1.7	Ours in Section 5.1
	19	$2^{19.70}$	$2^{16.58}$	0.18	1.4	Ours in Section 5.1
SIMON96	14	$2^{19.16}$	$2^{16.32}$	0.35	3.3	Ours in Section 5.2
SIMON64	13	$2^{25.70}$	$2^{13.24}$	0.57	–	Hou et al. (2023)
	13	$2^{18.00}$	$2^{15.58}$	0.94	1.1	Ours in Section 5.3
	14	$2^{18.37}$	$2^{15.58}$	0.44	1.6	Ours in Section 5.3
SPECK128	12	$2^{23.30}$	$2^{16.32}$	0.52	3.2	Chen et al. (2022)
	12	$2^{22.96}$	$2^{16.32}$	0.70	1.4	Ours in Section 5.4
SPECK96	10	$2^{20.94}$	$2^{16.00}$	0.81	1.4	Chen et al. (2022)
	10	$2^{19.92}$	$2^{16.00}$	0.83	0.4	Ours in Section 5.4
SPECK64	9	$2^{18.13}$	$2^{15.58}$	0.90	1.6	Chen et al. (2022)
	9	$2^{18.10}$	$2^{15.58}$	0.96	0.6	Ours in Section 5.4

hw(kg, rk) is the average Hamming distance between the guessed key and the round key. "–" means that Hou et al. (2023) used the parameter hw(kg, rk) ≤ 7 to calculate the success rate of key recovery. However, they did not disclose the specific value of hw(kg, rk). We use the parameter hw(kg, rk) ≤ 3 to calculate the success rate of key recovery

SIMON can handle several block sizes and key sizes. Its parameters are provided in the supplementary materials.

SIMON uses the Feistel structure, with its round transformation comprising primarily of three operations: bitwise AND &, bitwise XOR $\oplus$, and circular shift. The iterative process of the round function is expressed as follows:

$$\begin{cases} L_{i+1} = F\left(L_i\right) \oplus R_i \oplus \mathrm{rk}_i, \\ R_{i+1} = L_i, \end{cases} \tag{1}$$

where $F\left(x\right) = \left(\left(S^8 x\right) \,\&\, \left(S^1 x\right)\right) \oplus \left(S^2 x\right)$. The round transformation of SIMON is shown in the supplementary materials.

The SIMON key expansion algorithm is determined by the initial number of keywords and the block size. Considering SIMON$2n/mn$, its seed key comprises subkeys from previous m rounds, indicated as K, while the remaining subkeys are produced by the following equation:

$$\mathrm{rk}_{i+m} = \begin{cases} c \oplus \left(z_j\right)_i \oplus \mathrm{rk}_i \oplus \left(I \oplus S^{-1}\right) S^{-3}\mathrm{rk}_{i+1}, \\ \qquad\qquad \text{if } m = 2, \\ c \oplus \left(z_j\right)_i \oplus \mathrm{rk}_i \oplus \left(I \oplus S^{-1}\right) S^{-3}\mathrm{rk}_{i+2}, \\ \qquad\qquad \text{if } m = 3, \\ c \oplus \left(z_j\right)_i \oplus \mathrm{rk}_i \oplus \left(I \oplus S^{-1}\right) \\ \cdot \left(S^{-3}\mathrm{rk}_{i+3} \oplus \mathrm{rk}_{i+1}\right), \text{ if } m = 4, \end{cases} \tag{2}$$

where $c = 2^n - 4$ and $\left(z_j\right)_i$ is the i^{th} bit of z_j.

Unlike SIMON, SPECK (Beaulieu et al., 2015) is intended to function as a lightweight block cipher for software that operates under conditions where resources are constrained. Moreover, the algorithm ensures that flexibility is given importance, and consequently, SPECK offers support for a range of block sizes and key sizes. These parameters are provided in the supplementary materials.

SPECK uses the ARX structure, with its round transformation centered on three operations: addition modulo 2^n $\odot$, bitwise XOR $\oplus$, and circular shift. The iterative process of the round transformation can be represented as follows:

$$\begin{cases} L_{i+1} = \left(\left(S^{-\alpha} L_i\right) \odot R_i\right) \oplus \mathrm{rk}_i, \\ R_{i+1} = \left(S^{\beta} R_i\right) \oplus L_{i+1}, \end{cases} \tag{3}$$

where α and β are circular shift parameters. The round transformation of SPECK is shown in the supplementary materials.

The key expansion algorithm of SPECK is related to the number of words of its initial key. In the case of SPECK$2n/mn$, the seed key K comprises a total of m words, which are denoted as $(l_{m-2}, l_{m-1}, \cdots, l_1, \mathrm{rk}_0)$, and the subsequent subkeys are produced accordingly:

$$\begin{cases} l_{i+m-1} = \left(\mathrm{rk}_i \odot S^{-\alpha} l_i\right) \oplus i, \\ \mathrm{rk}_{i+1} = S^{\beta}\mathrm{rk}_i \oplus l_{i+m-1}. \end{cases} \tag{4}$$

The SPECK and SIMON algorithms, after their publication, have held a significant position among lightweight block algorithms, making it crucial to analyze their security. In addition, due to the rapid development of big data and big models, the scale of data is increasing, resulting in a growing demand for encryption. Large-state block ciphers are capable of meeting this demand, and their application is becoming increasingly widespread. Therefore, it is particularly important to conduct security analysis of large-state block ciphers, especially the SPECK and SIMON algorithms.

2.3 Gohr's neural distinguisher and key recovery attack

Gohr (2019) proposed a neural network distinguisher for SPECK$32/64$, which was built on a deep residual network. Each sample in the deep residual network's training data comprises a ciphertext pair and a label, wherein a ciphertext pair with a label of 1 indicates a real ciphertext pair, the difference between the corresponding plaintext pairs of such a ciphertext pair is Δ, a ciphertext pair with a label of 0 indicates a random ciphertext pair, and the difference between the corresponding plaintext pairs of such a ciphertext pair is a random value. To effectively train a neural network, it is necessary to possess both a training set and a validation set. In each set, 50% of the samples should have a label of 1 and the remaining 50% should have a label of 0.

This study uses a fundamental training framework which involves training a deep residual network in 50 epochs, each epoch consisting of 1×10^7 samples from the training set. During each epoch, the dataset is divided into groups, each with a batch size of 5000 samples. The error function of the deep residual network is the sum of the mean square error loss and the L_2 weight regularization term, and the regularization factor is 1×10^{-5}. The optimization method is the Adam algorithm provided by

Keras (Kingma and Ba, 2017). The learning rate $l_i = a + \frac{(n-i) \bmod (n+1)}{n}(b-a)$ of each epoch i is determined using the cyclic learning rate framework, $a = 1 \times 10^{-4},\ b = 2 \times 10^{-3},$ and $n = 9$.

The base version of Gohr (2019)'s key recovery attack is given an r_1-round pre-differential $\Gamma \xrightarrow{p} \Delta$ and an r_2-round neural network distinguisher ND, recovering the $(r_1 + r_2 + 1)$-round key. Gohr's key recovery attack process is shown in the supplementary materials. Neutral bits of the r_1-round pre-differential are used to expand each data pair. $(P, P+\Gamma)$ is pair of plaintext. A short r_1-round pre-differential $\Gamma \xrightarrow{p} \Delta$ with probability denoted by p is prepended on top of the hybrid distinguisher to increase the number of rounds of the key recovery attack. To enhance the deep learning based key recovery attack, Gohr combined Bayesian optimization to execute the key recovery attack on 11- and 12-round SPECK32/64.

2.4 Multi-stage deep learning aided key recovery framework

As the neural distinguisher takes a complete ciphertext pair as the input, it is required to guess all the bits of the round key at the same time in a key recovery attack, which is obviously difficult for large-state block ciphers. Therefore, Chen et al. (2023) proposed the concepts of information bit, teacher distinguisher, and student distinguisher, and introduced the idea of building a student distinguisher on some ciphertext bits to reduce the key space:

Definition 1 (Chen et al., 2023) For a cipher reduced to h rounds, the neural distinguisher trained on the complete ciphertexts (C_0, C_1) is denoted as the teacher distinguisher $\mathrm{ND}_h^{\mathrm{t}}$, and the neural distinguisher trained on partial ciphertext bits $(\varphi(C_0, \Gamma), \varphi(C_1, \Gamma))$ is denoted as the student distinguisher $\mathrm{ND}_h^{\mathrm{s}}$. The teacher distinguisher is viewed as a special student distinguisher.

According to Definition 1, identifying information bits is challenging, meaning that the creation of a student neural distinguisher should be based on them. Chen et al. (2023) introduced the concept of information bits for a neural distinguisher:

Definition 2 (Chen et al., 2023) For $\mathrm{ND}_h^{\mathrm{t}}$, if the distinguishing accuracy is significantly affected by the j^{th} bit of C_0 or C_1, the j^{th} ciphertext bit is an informative bit.

After giving the above definition, Chen et al. (2023) proposed the multi-stage deep learning aided framework. The key recovery attack in the multi-stage deep learning aided framework is divided into several stages. Each stage incorporates an independent neural distinguisher to recover specific portions of the key bits, and a neural distinguisher combination is formed by the neural distinguishers in each stage. At every stage, we implement Gohr's fundamental attack version and may apply the accelerated version of Gohr's attack.

Definition 3 (Chen et al., 2023) An informative bit is the ciphertext bit that is helpful to distinguish between the cipher and a pseudo-random permutation.

Assuming that the attacker has selected a neural distinguisher combination of x neural distinguishers and x pre-differentials $\mathrm{CD}_i := \Gamma_i \rightarrow \Delta_i$, where the probability of each pre-differential is $p_i, i \in \{1, 2, \cdots, x\}$, and the aim of the attack is to recover the last round key rk.

The attack consists of x stages. During stage i, rk's $|B_i|$ key bits rely on the findings from the previous $i-1$ stages. Therefore, in stage i, the knowledge of $\bigcup_{j \in \{1,2,\cdots,i-1\}} B_j$ is already established, enabling the recovery of a section of the key bit set B_i via either the basic or accelerated versions of the Gohr attack. Here, we use the basic version of the attack for every stage, whereby every stage is associated with a neural distinguisher and recovers $|B_i|$ key bits, $i \in \{1, 2, \cdots, x\}$. The complete process of attack is demonstrated in the supplementary materials (Bao et al., 2022).

3 Improved deep learning aided key recovery framework and applications to large-state block ciphers

In this section, we investigate and enhance the deep learning aided key recovery framework for large-state block ciphers proposed by Chen et al. (2022). The selection of neural distinguisher combinations greatly influences the results of the key recovery attack. Therefore, we suggest using the key bit sensitivity test (KBST) to devise a novel approach for developing neural distinguisher combinations that can enhance the deep learning aided key recovery system for large-state block ciphers. In this section, we first introduce the KBST technique. We then propose a new method to construct neural distinguisher

combinations based on the test results of the KBST technique. This will improve the deep learning aided key recovery framework for large-state block ciphers. Following this, we provide an explanation of how the multi-stage key recovery framework can be improved from the perspective of cryptanalysis. Finally, we discuss the complexity of key recovery attacks.

3.1 KBST technique

Building upon Chen et al. (2023), we propose the KBST technique, which considers the direct influence of the key on the ciphertext state. By dividing the round key of the last round using this technique and combining it with the structure of the algorithm round function, we can identify the ciphertext bit linked to the key sensitive bit, and subsequently develop a student distinguisher for that specific ciphertext bit. The KBST technique is provided in the supplementary materials.

In the supplementary materials, we show the outcomes of KBST on the neural distinguisher for 15-round SIMON128. The input difference of the neural distinguisher for the 15-round SIMON128 teachers is $(0x0, 0x200000000)$. The results identify that the neural distinguisher accuracy is predominantly affected by key bits 14–33.

Using the KBST technique, we partition the entire round key into various subkey spaces. In Section 3.2, we apply the results of KBST to enhance the multi-stage key recovery framework.

3.2 Improved multi-stage deep learning aided key recovery framework

Building on the multi-stage key recovery framework proposed by Chen et al. (2022), we enhance the framework by creating appropriate neural distinguisher combinations using the KBST technique described in Section 3.1. In Section 4, we will present the neural distinguisher combinations constructed for different versions of large-state SIMON and SPECK.

In a multi-stage deep learning aided key recovery framework, discovering an appropriate neural distinguisher combination is vital to a successful key recovery attack. Therefore, we suggest the following technique for constructing a suitable combination of neural distinguishers:

1. Train ω teacher neural distinguishers, each using a different plaintext difference $\Delta_{[i]}$. Additionally, ω denotes the block size of the plaintext. The inputs to each of these ω neural distinguishers are complete ciphertext pairs, with a total of 2ω bits. For SIMON2n and SPECK2n, ω equals $2n$.

2. Detecting the information bits that correspond to each teacher neural distinguisher involves using KBST. In this process, we identify the key bits that have an impact on the accuracy of the teacher neural distinguisher. Next, we locate the partial state bits that align with the key bits in accordance with the round function structure of the algorithm. The set K_i denotes the collection of sensitive key bits, while the set C_i denotes the collection of partial state bits, $i \in \{0, 1, \cdots, \omega - 1]\}$.

3. Based on K_i and $C_i, i \in \{0, 1, \cdots, \omega - 1\}$, we select the neural distinguisher combination. The neural distinguisher combination comprises x student neural distinguishers. These x student neural distinguishers are generated based on the following criteria:

(1) Choose x teacher neural distinguishers and ensure that their accuracies are high. The accuracy of the corresponding teacher neural distinguisher may vary depending on different input differences $\Delta_i, i \in \{0, 1, \cdots, \omega - 1\}$. Therefore, it is necessary to select the teacher neural distinguisher with higher accuracy. Furthermore, it is imperative for the student neural distinguisher used in the key recovery to possess an accuracy of at least 65%, or even higher. Therefore, it is essential for the x teacher neural distinguishers selected to have a relatively high accuracy.

(2) When training x student neural distinguishers, the input selection of each distinguisher is determined by the partial state bits that correspond to the key bits. According to the result of KBST and the algorithm's key expansion algorithm, it is advisable to choose input bits for each student neural distinguisher from the partial ciphertext state bits that correspond to the key bits sensitive to the neural distinguisher. This will maximize the accuracy of the trained student distinguisher. A comprehensive theoretical analysis of this approach will be provided in Section 3.3. For the newly trained distinguisher of SIMON2n and SPECK2n, a partial ciphertext of length $4\,|C_i|$ serves as the input.

(3) The selection of key bits recovered by each neural distinguisher should be based on the

significance of their impact on the accuracy of the student neural distinguisher. The concatenation $B_1 \cup B_2 \cdots \cup B_x$ of the set $B_i, i \in \{1, 2, \cdots, x\}$, consisting of the key bits recovered by x neural distinguishers, is expected to include almost all, if not all, of the final round key bits.

(4) After selecting x teacher neural distinguishers along with their corresponding partial state bits and recovered key bits, the partial ciphertext state bits are used to train the student distinguishers. As a result, x student neural distinguishers are obtained, which collectively form a neural distinguisher combination.

4. At this stage, we have identified a neural distinguisher combination, comprising student neural distinguishers. This guarantees the highest achievable accuracy for the student neural distinguisher constructed using partial ciphertext state bits, while ensuring that the task of recovering partial keys for each student neural distinguisher is both manageable and equitable.

Using the above method of constructing neural distinguishers, we construct a combination of neural distinguishers to improve the multi-stage key recovery framework. The improved key recovery attack flowchart is demonstrated in the supplementary materials.

3.3 Interpretation from the cryptanalysis perspective

When building a neural distinguisher combination, it is necessary to perform KBST for each teacher neural distinguisher, choosing suitable partial ciphertext bits to construct the student neural distinguisher and determining which key bits require recovery. In this subsection, we examine the test outcomes from the perspective of cryptanalysis and establish the theoretical framework for constructing the neural distinguisher combination.

According to the outcomes of KBST, we categorize the sensitive key bits into two groups and provide the subsequent definition:

Definition 4 The key bits that significantly affect the neural distinguisher are referred to as true impact key bits, whereas key bits that have a minimal impact on the neural distinguisher are referred to as pseudo impact key bits.

Based on our observation of KBST and the algorithm key expansion approach, we determine that heightened sensitivity of a specific key bit in the neural distinguisher may affect the sensitivity of other key bits. In the subsequent section, we will provide an explanation and analysis of this phenomenon in relation to related keys.

Knudsen (1991) and Biham (1994) proposed the related key attack, which reflects how key expansion algorithms affect the security of block ciphers. Key expansion algorithms expand a seed key into multiple subkeys, and then embed each subkey into the encryption (decryption) algorithm round transformation. During a related key attack, the attacker is unable to discern the key, but can identify other keys linked to the current one. The attacker carefully selects the relationship between these keys and capitalizes on the weaknesses of the block cipher key expansion. Ultimately, the attacker seeks to establish a correlation between the original key and the encryption algorithms associated with the newly identified related key, all with the goal of recovering the key.

By analyzing the key expansion algorithms of SIMON and SPECK, along with the related key attack ideas, we investigate the correlation between the last and second-to-last round keys generated by the key expansion scheme. Additionally, we scrutinize the results for KBST.

For demonstrating the SIMON algorithm, we use SIMON64/96 as an example. The key expansion algorithm for SIMON64/96 is outlined with $\mathrm{rk}_{i+3} = c \oplus (z_j)_i \oplus \mathrm{rk}_i \oplus \left(I \oplus S^{-1}\right) S^{-3}\mathrm{rk}_{i+2}$. We analyze the correlation between the 14- and 13-round keys.

In Table 2, if bit 0 of rk_{14} is determined to be the true impact key bit (i.e., more sensitive), it corresponds to bits 3 and 4 of rk_{13}, according to the correlation among rk_{14}, $S^{-3}\mathrm{rk}_{13}$, and $S^{-1}(S^{-3}\mathrm{rk}_{13})$. Bits 3 and 4 of rk_{13} have an effect on bits 1 and 31 of rk_{14}, which means that if bit 1 or 31 of rk_{14} is less sensitive, then it is likely that the sensitivity of bit 1 or 31 of rk_{14} is caused by bit 0 of rk_{14}.

For demonstrating the SPECK algorithm, we use SPECK64/96 as an example. The key expansion algorithm for SPECK64/96 is outlined below:

$$\begin{cases} l_{i+m-1} = (\mathrm{rk}_i \odot S^{-\alpha} l_i) \oplus i, \\ \mathrm{rk}_{i+1} = S^{\beta}\mathrm{rk}_i \oplus l_{i+m-1}. \end{cases} \tag{5}$$

We will analyze the correlation between the 9- and 8-round keys.

In Table 3, if bit 0 of rk_9 is determined to be the true impact key bit (i.e., more sensitive), it

corresponds to bits 0 and 29 of rk_8, according to the correlation among rk_9, rk_8, and $S^3\mathrm{rk}_8$. Bits 0 and 29 of rk_8 have an effect on bits 3 and 29 of rk_9, which means that if bit 3 or 29 of rk_9 is less sensitive, then it is likely that the sensitivity of bit 3 or 29 of rk_9 is caused by bit 0 of rk_9.

The experimental results concur with the theoretical analysis. As a result, we aim to select partial ciphertext state bits that relate to the true impact key bits and the pseudo impact key bits while selecting the input of the student neural distinguisher. This selection should be as comprehensive as possible, focusing on the key bits that are true impact key bits recovered by the neural distinguisher. It helps minimize the Hamming distance between the returned key guess and the actual key. This aforementioned attribute is also valid for other versions of the SIMON and SPECK algorithms.

3.4 Complexity analysis

We explore the maximum complexity of the attack, specifically its complexity in the worst case scenario.

In the most unfavorable scenario, each phase uses $\frac{\varepsilon}{p_i}$ ciphertext structures and $\log_2 N_i$ neutral bits, $i \in \{1, 2, \cdots, x\}$. Therefore, the maximum limit on the data complexity of the attack is $\sum_{i=1}^{x} \frac{\varepsilon}{p_i} N_i$.

In the attack, a ciphertext pair is decrypted using a key guess. The decrypted pseudo-ciphertext pair is then fed to the neural distinguisher, and the prediction score G is obtained, followed by computation of $\log_2 \frac{G}{1-G}$. Considering that this operation is considered basic, the attack performs it $\sum_{i=1}^{x} \beta_i 2^{|\beta_i|} \frac{\varepsilon}{p_i} N_i$ times. If one equates a basic operation with δ_i encryptions of the target algorithm, $i \in \{1, 2, \cdots, x\}$, the upper bound on the computational complexity (i.e., the number of encryptions) of the attack is $\sum_{i=1}^{x} \beta_i 2^{|\beta_i|} \frac{\varepsilon}{p_i} N_i \delta_i$. Moreover, by varying the number of neutral bits, one can regulate the data and time complexity of the multi-stage key recovery attack.

4 Neural distinguishers for large-state SIMON and SPECK

We use the method of constructing neural distinguisher combinations proposed in Section 3 to find an effective combination of neural distinguishers for every large-state member of SIMON and SPECK. To clarify, the key sizes of SIMON and SPECK do not affect the neural distinguishers. Therefore, we refer to members of SPECK and SIMON with different key sizes but equivalent block size of $2n$ bits as SPECK2n and SIMON2n, respectively.

4.1 Neural distinguishers for large-state SIMON

Using the proposed methodology for constructing neural distinguishers as detailed in Section 3,

Table 2 The correlation between the 14- and 13-round keys of SIMON64/96

Key	Position of the key bit															
rk_{14}	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
$S^{-3}\mathrm{rk}_{13}$	2 18	1 17	0 16	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3
$S^{-1}(S^{-3}\mathrm{rk}_{13})$	3 19	2 18	1 17	0 16	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4

Table 3 The correlation between the 9- and 8-round keys of SPECK64/96

Key	Position of the key bit															
rk_9	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
rk_8	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
$S^3\mathrm{rk}_8$	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	15 31	14 30	13 29

we develop 15- and 16-round neural distinguisher combinations for SIMON128, 11-round neural distinguisher combination for SIMON96, and 10- and 11-round neural distinguisher combinations for SIMON64.

The 15-round SIMON128 neural distinguisher combination comprises six 15-round neural distinguishers, as illustrated in Table 4. These distinguishers are acquired from the following plaintext difference training:

$$\begin{aligned}&\Delta_1=\Delta_{[8]},\Delta_2=\Delta_{[22]},\Delta_3=\Delta_{[33]},\\&\Delta_4=\Delta_{[46]},\Delta_5=\Delta_{[55]},\Delta_6=\Delta_{[63]}.\end{aligned}\tag{6}$$

The 16-round SIMON128 neural distinguisher combination comprises six 16-round neural distinguishers, as illustrated in Table 4. These distinguishers are acquired from the following plaintext difference training:

$$\begin{aligned}&\Delta_1=\Delta_{[8]},\Delta_2=\Delta_{[22]},\Delta_3=\Delta_{[33]},\\&\Delta_4=\Delta_{[46]},\Delta_5=\Delta_{[54]},\Delta_6=\Delta_{[63]}.\end{aligned}\tag{7}$$

Table 4 outlines additional information regarding the 15- and 16-round neural distinguisher combinations for SIMON128, which have been used for practical key recovery attacks on 18- and 19-round SIMON128, correspondingly.

The 11-round SIMON96 neural distinguisher combination comprises five 11-round neural distinguishers, as illustrated in Table 5. These distinguishers are acquired from the following plaintext difference training:

$$\begin{aligned}&\Delta_1=\Delta_{[12]},\Delta_2=\Delta_{[20]},\Delta_3=\Delta_{[27]},\\&\Delta_4=\Delta_{[38]},\Delta_5=\Delta_{[47]}.\end{aligned}\tag{8}$$

Table 5 outlines additional information regarding the 11-round neural distinguisher combination for SIMON96, which is used to execute a practical key recovery attack on 14-round SIMON96.

Table 5 11-round neural distinguisher combination for SIMON96

ND_i	Δ_i	B_i	C_i	Accuracy
ND_1	$\Delta_{[12]}$	$\{8\sim 0\}$	$\{14\sim 0\}$	0.610
ND_2	$\Delta_{[20]}$	$\{19\sim 9\}$	$\{20\sim 9\}$	0.652
ND_3	$\Delta_{[27]}$	$\{27\sim 20\}$	$\{27\sim 16\}$	0.635
ND_4	$\Delta_{[38]}$	$\{38\sim 28\}$	$\{38\sim 24\}$	0.658
ND_5	$\Delta_{[47]}$	$\{47\sim 39\}$	$\{47\sim 39\}$	0.753

The 10-round SIMON64 neural distinguisher combination comprises three 10-round neural distinguishers, as illustrated in Table 6. These distinguishers are acquired from the following plaintext difference training:

$$\Delta_1=\Delta_{[10]},\Delta_2=\Delta_{[19]},\Delta_3=\Delta_{[30]}.\tag{9}$$

The 11-round SIMON64 neural distinguisher combination comprises three 11-round neural distinguishers, as illustrated in Table 6. These distinguishers are acquired from the following plaintext difference training:

$$\Delta_1=\Delta_{[10]},\Delta_2=\Delta_{[19]},\Delta_3=\Delta_{[27]}.\tag{10}$$

Table 6 outlines additional information regarding the 10- and 11-round neural distinguisher combinations for SIMON64, which have been used for practical key recovery attacks on 13- and 14-round SIMON64, correspondingly.

4.2 Neural distinguishers for large-state SPECK

The neural distinguisher combinations for the large-state SPECK are reconstructed using the method proposed in Section 3. Correspondingly, the inputs of the neural distinguishers are adjusted in the SPECK128, SPECK96, and SPECK64 neural distinguisher combinations. Further details summarizing the neural distinguisher combinations for the three

Table 4 15- and 16-round neural distinguisher combinations for SIMON128

ND_i	15-round SIMON128				16-round SIMON128			
	Δ_i	B_i	C_i	Accuracy	Δ_i	B_i	C_i	Accuracy
ND_1	$\Delta_{[8]}$	$\{8\sim 0\}$	$\{8\sim 0\}$	0.714	$\Delta_{[8]}$	$\{9\sim 0\}$	$\{9\sim 0\}$	0.630
ND_2	$\Delta_{[22]}$	$\{21\sim 9\}$	$\{21\sim 7\}$	0.770	$\Delta_{[22]}$	$\{21\sim 10\}$	$\{21\sim 7\}$	0.653
ND_3	$\Delta_{[33]}$	$\{32\sim 22\}$	$\{32\sim 18\}$	0.770	$\Delta_{[33]}$	$\{32\sim 22\}$	$\{32\sim 18\}$	0.653
ND_4	$\Delta_{[46]}$	$\{45\sim 33\}$	$\{45\sim 31\}$	0.770	$\Delta_{[46]}$	$\{45\sim 33\}$	$\{45\sim 31\}$	0.652
ND_5	$\Delta_{[55]}$	$\{54\sim 46\}$	$\{54\sim 40\}$	0.770	$\Delta_{[54]}$	$\{53\sim 46\}$	$\{53\sim 42\}$	0.640
ND_6	$\Delta_{[63]}$	$\{63\sim 55\}$	$\{63\sim 49\}$	0.783	$\Delta_{[63]}$	$\{63\sim 54\}$	$\{63\sim 52\}$	0.660

types of large-state SPECK are provided in the supplementary materials. Practical key recovery attacks on SPECK128, SPECK96, and SPECK64 have been enhanced with neural distinguisher combinations.

5 Practical key recovery attacks on large-state SIMON and SPECK

In this section, using the neural distinguisher combinations proposed in Section 4, we demonstrate the effectiveness of our enhanced multi-stage key recovery framework through the practical key recovery attack on large-state SIMON and SPECK. For all attacks, we define a successful attack as $\mathrm{hw}(\mathrm{kg}, \mathrm{rk}) \leq 3$.

5.1 Practical key recovery attacks on 18- and 19-round SIMON128

5.1.1 Practical key recovery attack on 18-round SIMON128

By using the enhanced framework for multi-stage key recovery proposed in Section 3, in combination with the neural distinguisher combination constructed in Section 4, the last round of complete subkeys can be recovered. Specifically, each of the six 15-round neural distinguishers (refer to Table 4) is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 7) to create a 16-round hybrid distinguisher HD_i. Since the initial non-linear SIMON operation is not whitened beforehand, it is feasible to elongate HD_i by one round atop every 16 rounds without any additional cost. Consequently, there are a total of 17 rounds in addition to the last round, with the objective of the attack being to retrieve the 18-round key rk_{18}. The attack parameters are displayed in Tables 4 and 7.

Table 7 presents a summary of the six one-round pre-differentials used, along with their corresponding difference probabilities p_i and 10 neutral bits ($N_i = 2^{10}$, $i \in \{1, 2, \cdots, 6\}$). These values are obtained by setting $\epsilon = 4$, which ensures that stage i produces a maximum of $\frac{4}{p_i}$ ciphertext structures, $i \in \{1, 2, \cdots, 6\}$. The threshold for filtering out erroneous keys during the six stages is denoted by $c_i = 10$, $i \in \{1, 2, \cdots, 6\}$. Furthermore, the number of surviving keys retained in stage i, $i \in \{2, 3, \cdots, 6\}$, for the subsequent stage is bounded above by $\beta_i = 3$ (where β_1 always equals 1). Using the given attack parameters, we conduct 100 trials on a computer equipped with a single graphics card (Geforce RTX 3090 GPU-equipped computer). The average time spent in executing the aforementioned attack with one graphics card and one CPU core is 492 s.

Out of 100 trials, a total of 91 trails return a final key kg. The Hamming distance between the guessed key kg and the correct round key rk is indicated as $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$. In the aforementioned 91 trials, the average Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ is 1.7. Detailed statistics are presented in Table 8. We define a successful attack as $\mathrm{hw}(\mathrm{kg}, \mathrm{rk}) \leq 3$. According to Table 8, the success rate of the 18-round key recovery attack is 81%.

Table 6 10- and 11-round neural distinguisher combinations for SIMON64

ND_i	10-round SIMON64				11-round SIMON64			
	Δ_i	B_i	C_i	Accuracy	Δ_i	B_i	C_i	Accuracy
ND_1	$\Delta_{[10]}$	$\{11 \sim 0\}$	$\{11 \sim 0\}$	0.662	$\Delta_{[10]}$	$\{10 \sim 0\}$	$\{10 \sim 0\}$	0.586
ND_2	$\Delta_{[19]}$	$\{20 \sim 12\}$	$\{20 \sim 9\}$	0.670	$\Delta_{[19]}$	$\{19 \sim 11\}$	$\{19 \sim 5\}$	0.599
ND_3	$\Delta_{[30]}$	$\{31 \sim 21\}$	$\{31 \sim 17\}$	0.711	$\Delta_{[27]}$	$\{31 \sim 20\}$	$\{31 \sim 17\}$	0.606

Table 7 One-round pre-differentials added before the neural distinguishers in Table 4

CD_i	One-round SIMON128 (15-round ND)			One-round SIMON128 (16-round ND)		
	$\Gamma_i \to \Delta_i$	Neutral bit	p_i	$\Gamma_i \to \Delta_i$	Neutral bit	p_i
CD_1	$\Delta_{[72,16,10,9]} \to \Delta_{[8]}$	$\{[20] \sim [11]\}$	2^{-2}	$\Delta_{[72,16,10,9]} \to \Delta_{[8]}$	$\{[20] \sim [11]\}$	2^{-2}
CD_2	$\Delta_{[86,30,24,23]} \to \Delta_{[22]}$	$\{[32] \sim [23]\}$	2^{-2}	$\Delta_{[86,30,24,23]} \to \Delta_{[22]}$	$\{[32] \sim [23]\}$	2^{-2}
CD_3	$\Delta_{[97,41,35,34]} \to \Delta_{[33]}$	$\{[46] \sim [37]\}$	2^{-2}	$\Delta_{[97,41,35,34]} \to \Delta_{[33]}$	$\{[46] \sim [37]\}$	2^{-2}
CD_4	$\Delta_{[110,54,48,47]} \to \Delta_{[46]}$	$\{[61] \sim [52]\}$	2^{-2}	$\Delta_{[110,54,48,47]} \to \Delta_{[46]}$	$\{[61] \sim [52]\}$	2^{-2}
CD_5	$\Delta_{[119,63,57,56]} \to \Delta_{[55]}$	$\{[73] \sim [64]\}$	2^{-2}	$\Delta_{[118,62,56,55]} \to \Delta_{[54]}$	$\{[73] \sim [64]\}$	2^{-2}
CD_6	$\Delta_{[127,7,1,0]} \to \Delta_{[63]}$	$\{[83] \sim [74]\}$	2^{-2}	$\Delta_{[127,7,1,0]} \to \Delta_{[63]}$	$\{[83] \sim [74]\}$	2^{-2}

Table 8 Statistics on hw(kg, rk) over 100 trials of an 18-round attack on SIMON128

hw(kg, rk)	Number of trails	hw(kg, rk)	Number of trails
0	20	3	16
1	23	4	9
2	22	5	1

5.1.2 Practical key recovery attack on 19-round SIMON128

The general method of attack is similar to the 18-round attack. Specifically, each of the six 16-round neural distinguishers (refer to Table 4) is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 7) to create a 17-round hybrid distinguisher HD_i. One additional round of expansion is available before each 17-round HD_i. Consequently, there are a total of 18 rounds in addition to the last round, with the objective of the attack being to recover the 19-round key rk_{19}. The attack parameters are displayed in Tables 4 and 7.

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 416 s.

Out of 100 trials, a total of 18 trails return a final key kg. In the aforementioned 18 trials, the average Hamming distance hw(kg, rk) is 1.4. Detailed statistics are presented in Table 9. The success rate of the 19-round key recovery attack is 18%.

5.2 Practical key recovery attack on 14-round SIMON96

By using the enhanced framework for multi-stage key recovery proposed in Section 3, in combination with the neural distinguisher combination constructed in Section 4, the last round of complete subkeys can be recovered. Specifically, each of the five 11-round neural distinguishers (refer to Table 5) is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 10) to create a 12-round hybrid distinguisher HD_i. One additional round of expansion is available before each 12-round HD_i. Consequently, there are a total of 13 rounds in addition to the last round, with the objective of the attack being to recover the 14-round key rk_{14}. The attack parameters are displayed in Tables 5 and 10.

Table 9 Statistics on hw(kg, rk) over 100 trials of a 19-round attack on SIMON128

hw(kg, rk)	Number of trails	hw(kg, rk)	Number of trails
0	1	3	1
1	9	4	0
2	7	5	0

Table 10 One-round pre-differentials added before the neural distinguishers in Table 5

CD_i	$\Gamma_i \to \Delta_i$	Neutral bit	p_i
CD_1	$\Delta_{[60,20,14,13]} \to \Delta_{[12]}$	$\{[25] \sim [16]\}$	2^{-2}
CD_2	$\Delta_{[68,28,22,21]} \to \Delta_{[20]}$	$\{[37] \sim [28]\}$	2^{-2}
CD_3	$\Delta_{[75,35,29,28]} \to \Delta_{[27]}$	$\{[49] \sim [40]\}$	2^{-2}
CD_4	$\Delta_{[86,46,40,39]} \to \Delta_{[38]}$	$\{[61] \sim [52]\}$	2^{-2}
CD_5	$\Delta_{[95,7,1,0]} \to \Delta_{[47]}$	$\{[71] \sim [62]\}$	2^{-2}

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 229 s.

Out of 100 trials, a total of 56 trails return a final key kg. In the aforementioned 56 trials, the average Hamming distance hw(kg, rk) is 3.3. Detailed statistics are presented in Table 11. The success rate of the 14-round key recovery attack is 35%.

Table 11 Statistics on hw(kg, rk) over 100 trials of a 14-round attack on SIMON96

hw(kg, rk)	Number of trails	hw(kg, rk)	Number of trails
0	1	5	7
1	8	6	4
2	12	7	3
3	14	8	1
4	6		

5.3 Practical key recovery attacks on 13- and 14-round SIMON64

5.3.1 Practical key recovery attack on 13-round SIMON64

Each of the three 10-round neural distinguishers (refer to Table 6) is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 12) to create a 11-round hybrid distinguisher HD_i. One additional round of expansion is available before each 11-round HD_i. Consequently, there are a total of 12 rounds in addition to the last round, with the objective of the attack being to recover the 13-round key rk_{13}. The attack parameters are displayed in Tables 6 and 12.

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 54 s.

Out of 100 trials, a total of 95 trails return a final key kg. In the aforementioned 95 trials, the average Hamming distance hw(kg, rk) is 1.0. Detailed statistics are presented in Table 13. The success rate of the 13-round key recovery attack is 94%.

Compared to the attack proposed by Hou et al. (2023), the success rate of the enhanced key recovery attack on 13-round SIMON64 has risen from 57% to 94%, and the Hamming distance hw(kg, rk) between the guessed key kg and the correct round key rk and the time complexity decrease significantly.

5.3.2 Practical key recovery attack on 14-round SIMON64

The general method of attack is similar to the 13-round attack. Specifically, each of the three 11-round neural distinguishers (refer to Table 6) is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 12) to create a 12-round hybrid distinguisher HD_i. One additional round of expansion is available before each 12-round HD_i. Consequently, there are a total of 13 rounds in addition to the last round, with the objective of the attack being to retrieve the 14-round key rk_{14}. The attack parameters are displayed in Tables 6 and 12.

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 353 s.

Out of 100 trials, a total of 45 trails return a final key kg. In the aforementioned 45 trials, the average Hamming distance hw(kg, rk) is 1.6. Detailed statistics are presented in Table 14. The success rate of the 14-round key recovery attack is 44%.

5.4 Improvements to practical key recovery attacks on large-state SPECK

By using our advanced multi-stage key recovery framework as suggested in Section 3, we improve the actual key recovery for large-state SPECK. This once again confirms the validity of our improved multi-stage key retrieval framework.

5.4.1 Improvements to practical key recovery attack on 12-round SPECK128

The attack has been successfully carried out using the enhanced multi-stage key recovery framework as recommended in Section 3 and the novel 9-round neural distinguisher combination devised for SPECK128 as described in Section 4. The neural distinguisher combination uses five 9-round neural distinguishers. Each of the five 9-round neural distinguishers is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 15) to create a 10-round hybrid distinguisher HD_i. One additional round of expansion is available before each 10-round HD_i. Consequently, there are a total of 11 rounds in addition to the last round, with the objective of the attack being to recover the 12-round key rk_{12}.

Except for the different combination of neural distinguishers, the attack setup is identical to that of Chen et al. (2022).

Table 13 Statistics on hw(kg, rk) over 100 trials of a 13-round attack on SIMON64

hw(kg, rk)	Number of trails	hw(kg, rk)	Number of trails
0	30	3	7
1	39	4	1
2	18	5	0

Table 14 Statistics on hw(kg, rk) over 100 trials of a 14-round attack on SIMON64

hw(kg, rk)	Number of trails	hw(kg, rk)	Number of trails
0	8	3	9
1	14	4	1
2	13	5	0

Table 12 One-round pre-differentials added before the neural distinguishers in Table 6

CD_i	One-round SIMON64 (10-round ND)			One-round SIMON64 (11-round ND)		
	$\Gamma_i \to \Delta_i$	Neutral bit	p_i	$\Gamma_i \to \Delta_i$	Neutral bit	p_i
CD_1	$\Delta_{[42,18,12,11]} \to \Delta_{[10]}$	$\{[30] \sim [21]\}$	2^{-2}	$\Delta_{[40,16,10,9]} \to \Delta_{[8]}$	$\{[30] \sim [21]\}$	2^{-2}
CD_2	$\Delta_{[51,27,21,20]} \to \Delta_{[19]}$	$\{[38] \sim [29]\}$	2^{-2}	$\Delta_{[86,30,24,23]} \to \Delta_{[22]}$	$\{[32] \sim [23]\}$	2^{-2}
CD_3	$\Delta_{[62,31,6,0]} \to \Delta_{[30]}$	$\{[50] \sim [41]\}$	2^{-2}	$\Delta_{[59,29,28,3]} \to \Delta_{[33]}$	$\{[46] \sim [37]\}$	2^{-2}

Table 15 One-round pre-differentials added before the neural distinguishers in SPECK128, SPECK96, and SPECK64

CD_i	One-round SPECK128 (9-round ND)			One-round SPECK96 (7-round ND)			One-round SPECK64 (6-round ND)		
	$\Gamma_i \to \Delta_i$	Neutral bit	p_i	$\Gamma_i \to \Delta_i$	Neutral bit	p_i	$\Gamma_i \to \Delta_i$	Neutral bit	p_i
CD_1	$\Delta_{[72,69,61]} \to \Delta_{[64]}$	$\{[20] \sim [11]\}$	2^{-2}	$\Delta_{[61,58,2]} \to \Delta_{[53]}$	$\{[25] \sim [16]\}$	2^{-2}	$\Delta_{[50,47,7]} \to \Delta_{[42]}$	$\{[39] \sim [30]\}$	2^{-2}
CD_2	$\Delta_{[84,81,9]} \to \Delta_{[76]}$	$\{[32] \sim [23]\}$	2^{-2}	$\Delta_{[73,70,14]} \to \Delta_{[65]}$	$\{[37] \sim [28]\}$	2^{-2}	$\Delta_{[55,52,12]} \to \Delta_{[47]}$	$\{[39] \sim [30]\}$	2^{-2}
CD_3	$\Delta_{[98,95,23]} \to \Delta_{[90]}$	$\{[46] \sim [37]\}$	2^{-2}	$\Delta_{[85,82,26]} \to \Delta_{[77]}$	$\{[49] \sim [40]\}$	2^{-2}	$\Delta_{[41,38,30]} \to \Delta_{[33]}$	$\{[29] \sim [20]\}$	2^{-2}
CD_4	$\Delta_{[113,110,38]} \to \Delta_{[105]}$	$\{[61] \sim [52]\}$	2^{-2}	$\Delta_{[94,49,38]} \to \Delta_{[89]}$	$\{[61] \sim [52]\}$	2^{-2}			
CD_5	$\Delta_{[125,122,50]} \to \Delta_{[117]}$	$\{[73] \sim [64]\}$	2^{-2}						

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 3989 s.

Out of 100 trials, a total of 76 trails return a final key kg. In the aforementioned 76 trials, the average Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ is 1.4. Detailed statistics are presented in Table 16. We define a successful attack as $\mathrm{hw}(\mathrm{kg}, \mathrm{rk}) \leq 3$. According to Table 16, the success rate of the 12-round key recovery attack is 70%.

Compared to the attack proposed in Chen et al. (2022), the success rate of the enhanced key recovery attack on 12-round SPECK128 increases from 52% to 70%, and the Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ between the guessed key kg and the correct round key rk decreases from 3.2 to 1.4.

5.4.2 Improvements to practical key recovery attack on 10-round SPECK96

The attack has been successfully carried out using the enhanced multi-stage key recovery framework as recommended in Section 3 and the novel 7-round neural distinguisher combination devised for SPECK96 as described in Section 4. The neural distinguisher combination uses four 7-round neural distinguishers. Each of the four 7-round neural distinguishers is preceded by a one-round pre-differential $\Gamma_i \to \Delta_i$ (refer to Table 15) to create an 8-round hybrid distinguisher HD_i. One additional round of expansion is available before each 8-round HD_i. Consequently, there are a total of 9 rounds in addition to the last round, with the objective of the attack being to recover the 10-round key rk_{10}.

Expect for the different combination of neural distinguishers, the attack setup is identical to that of Chen et al. (2022).

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 407 s.

Out of 100 trials, a total of 83 trails return a final key kg. In the aforementioned 83 trials, the average Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ is 0.4. Detailed statistics are presented in Table 17. The success rate of the 10-round key recovery attack is 83%.

Compared to the attack proposed in Chen et al. (2022), the success rate of the enhanced key recovery attack on 10-round SPECK96 increases from 81% to 83%, and the Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ between the guessed key kg and the correct round key rk decreases from 1.4 to 0.4.

Table 16 Statistics on hw(kg, rk) over 100 trials of a 12-round attack on SPECK128

$\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$	Number of trails	
	Chen et al. (2022)	Ours in Section 5.4
0	4	29
1	12	12
2	17	18
3	19	11
4	8	5
5	9	0
6	7	1
7	3	0
8	0	0
9	1	0

Table 17 Statistics on hw(kg, rk) over 100 trials of a 10-round attack on SPECK96

$\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$	Number of trails	
	Chen et al. (2022)	Ours in Section 5.4
0	23	61
1	30	13
2	18	6
3	10	3
4	3	0
5	3	0

5.4.3 Improvements to practical key recovery attack on 9-round SPECK64

The attack has been successfully carried out using the enhanced multi-stage key recovery framework as recommended in Section 3 and the novel 6-round neural distinguisher combination devised for SPECK64 as described in Section 4. The neural distinguisher combination uses three 6-round neural distinguishers. Each of the three 6-round neural distinguishers is preceded by a one-round predifferential $\Gamma_i \to \Delta_i$ (refer to Table 15) to create a 7-round hybrid distinguisher HD_i. One additional round of expansion is available before each 7-round HD_i. Consequently, there are a total of 8 rounds in addition to the last round, with the objective of the attack being to recover the 9-round key rk_9. The attack parameters are displayed in Table 15 and the supplementary materials.

Except for the different combination of neural distinguishers, the attack setup is identical to that of Chen et al. (2022).

Under the aforementioned attack parameters, we conduct 100 experiments within the same experimental setup (i.e., a Geforce RTX 3090 GPU-equipped computer). The attack consumes an average of 88 s.

Out of 100 trials, a total of 96 trails return a final key kg. In the aforementioned 96 trials, the average Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ is 0.6. Detailed statistics are presented in Table 18. The success rate of the 9-round key recovery attack is 96%.

Compared to the attack proposed in Chen et al. (2022), the success rate of the enhanced key recovery attack on 9-round SPECK64 increases from 90% to 96%, and the Hamming distance $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ between the guessed key kg and the correct round key rk decreases from 1.6 to 0.6.

Table 18 Statistics on $\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$ over 100 trials of a 9-round attack on SPECK64

$\mathrm{hw}(\mathrm{kg}, \mathrm{rk})$	Number of trails	
	Chen et al. (2022)	Ours in Section 5.4
0	22	59
1	30	21
2	29	10
3	9	6
4	4	0
5	1	0
6	2	0
7	1	0

6 Conclusions

In this paper, we first propose a deep learning based KBST method to divide the key space, and study the effect of different single-bit input differences on the accuracy of large-state SIMON and SPECK neural distinguishers. Furthermore, we investigate the key bit sensitivities of diverse neural distinguishers using this method. Meanwhile, we propose a new technique for constructing neural distinguisher combinations. To confirm the validity of our KBST method and the efficiency of the neural distinguisher combination approach, we train a efficient neural distinguisher combination for every large-state member of SIMON and SPECK. Finally, we examine and enhance the deep learning aided key recovery framework for large-state block ciphers proposed by Chen et al. (2022) using the neural distinguisher combinations we developed. We conduct a practical key recovery attack on SIMON and SPECK large-state members.

To date attack on the 13-round SIMON64 and the attack on large-state SPECK presented in this paper are the most effective deep learning based practical key recovery attacks. Additionally, this work is the first to introduce deep learning based practical key recovery attacks on 18-round SIMON128, 19-round SIMON128, 14-round SIMON96, and 14-round SIMON64. This research advances the study of deep learning based key recovery attacks on large-state block ciphers and further promotes the research into the effects of deep learning on cryptanalysis.

In the subsequent phase, we shall delve deeper into the investigation of key recovery attacks based on deep learning for large-state block ciphers. This will include the construction of the neural distinguishers and key recovery of different block ciphers. Based on these studies, we aim to explore the intrinsic mechanism of the key recovery attack based on deep learning and provide a theoretical explanation. In addition, we will explore the possibility of integrating deep learning with other cryptanalysis techniques, e.g., exploring the development of linear distinguishers based on deep learning.

Contributors

Xiaowei LI designed the research, processed the data, and drafted the paper. Jiongjiong REN and Shaozhen CHEN

helped organize the paper. Xiaowei LI, Jiongjiong REN, and Shaozhen CHEN revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Baksi A, 2022. Machine learning-assisted differential distinguishers for lightweight ciphers. In: Baksi A (Ed.), Classical and Physical Security of Symmetric Key Cryptographic Algorithms. Springer, Singapore, p.141-162. https://doi.org/10.1007/978-981-16-6522-6_6

Bao ZZ, Guo J, Liu MC, et al., 2022. Enhancing differential-neural cryptanalysis. 28th Int Conf on the Theory and Application of Cryptology and Information Security, p.318-347. https://doi.org/10.1007/978-3-031-22963-3_11

Beaulieu R, Shors D, Smith J, et al., 2015. The SIMON and SPECK lightweight block ciphers. Proc 52nd Annual Design Automation Conf, Article 175. https://doi.org/10.1145/2744769.2747946

Bellini E, Rossi M, 2021. Performance comparison between deep learning-based and conventional cryptographic distinguishers. Proc Computing Conf on Intelligent Computing, p.681-701. https://doi.org/10.1007/978-3-030-80129-8_48

Biham E, 1994. New types of cryptanalytic attacks using related keys. *J Cryptol*, 7(4):229-246. https://doi.org/10.1007/BF00203965

Biham E, Shamir A, 1993. Differential cryptanalysis of the full 16-round DES. 12th Annual Int Cryptology Conf on Advances in Cryptology, p.487-496. https://doi.org/10.1007/3-540-48071-4_34

Chen Y, Yu HB, 2021. A new neural distinguisher model considering derived features from multiple ciphertext pairs. *Comput J*, Article 310. https://doi.org/10.1093/comjnl/bxac019

Chen Y, Bao ZZ, Shen YT, et al., 2022. A deep learning aided key recovery framework for large-state block ciphers. *Sci China Inform*, 53(7):1348-1367 (in Chinese). https://doi.org/10.1360/SSI-2022-0298

Chen Y, Shen YT, Yu HB, 2023. Neural-aided statistical attack for cryptanalysis. *Comput J*, 66(10):2480-2498. https://doi.org/10.1093/comjnl/bxac099

Gohr A, 2019. Improving attacks on round-reduced Speck32/64 using deep learning. 39th Annual Int Cryptology Conf on Advances in Cryptology, p.150-179. https://doi.org/10.1007/978-3-030-26951-7_6

Hou ZZ, Ren JJ, Chen SZ, 2023. Practical attacks of round-reduced SIMON based on deep learning. *Comput J*, 66(10):2517-2534. https://doi.org/10.1093/comjnl/bxac102

Jain A, Kohli V, Mishra G, 2020. Deep learning based differential distinguisher for lightweight cipher PRESENT. https://eprint.iacr.org/2020/846

Kingma DP, Ba J, 2017. Adam: a method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980

Knudsen LR, 1991. Cryptanalysis of LOKI. Int Conf on the Theory and Application of Cryptology, p.22-35. https://doi.org/10.1007/3-540-57332-1_2

Zhang L, Wang ZL, Wang BY, 2022. Improving differential-neural cryptanalysis with inception blocks. https://dblp.org/rec/journals/iacr/zhangWW22.html

List of supplementary materials

Algorithm S1 The multi-stage deep learning aided key recovery framework

Algorithm S2 Key bit sensitivity test

Table S1 The algorithm of Gohr's key recovery attack

Table S2 The parameters of the SIMON family

Table S3 The parameters of the SPECK family

Table S4 9-round neural distinguisher combination for SPECK128

Table S5 7-round neural distinguisher combination for SPECK96

Table S6 6-round neural distinguisher combination for SPECK64

Fig. S1 The round transformation of SIMON

Fig. S2 The round transformation of SPECK

Fig. S3 The outcomes of KBST on the neural distinguisher for 15-round SIMON128

Fig. S4 The schematic of the improved multi-stage key recovery framework for large-state block ciphers